NOSC

END
DATE
FILMED
11 8'
DTIC

NOSC

AD A105876

Technical Report 706

C2 RELEVANCE OF SELECTED ARTIFICIAL INTELLIGENCE SYSTEMS
Brief examination of several existing AI systems for use in future command and control applications

RJ Bechtel

7 July 1981

Interim Report for Period 1 October 1980 – 30 April 1981

Prepared for
Naval Electronic Systems Command
Code 613

**NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152**

81 10 20

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Report 706 (TR 706)	2. GOVT ACCESSION NO. AD-A105 876	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* C2 RELEVANCE OF SELECTED ARTIFICIAL INTELLIGENCE SYSTEMS Brief examination of several existing AI systems for use in future command and control applications		5. TYPE OF REPORT & PERIOD COVERED Interim 1 October 1980 – 30 April 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) RJ Bechtel		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62721N, F21241, XF21241100 (NOSC 824-CC96)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command, Code 613 Washington, DC 20360		12. REPORT DATE 7 July 1981
		13. NUMBER OF PAGES 24
14. MONITORING AGENCY NAME & ADDRESS(*If different from Controlling Office*)		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Artificial intelligence	Inference systems
Knowledge representation systems	Natural language processing
Knowledge presentation systems	Planning and problem solving

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A very brief examination of several existing artificial intelligence systems which have been judged to be worth further investigation for use in future command and control applications. This document represents the current state of an ongoing process which began with a much longer list of artificial intelligence systems. Each system on the long list was given a very quick evaluation on scales measuring availability, applicability, and maturity. Within each broad category of artificial intelligence effort (eg natural language, inference) the highest ranking systems were selected for further examination. This is the result of that examination.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73
S/N 0102-LF-014-6601

OBJECTIVE

Survey and evaluate existing artificial intelligence systems for applicability to command and control. Include assessments of availability and maturity of techniques.

RESULTS

Brief summaries of sixteen artificial intelligence systems are presented in five broad categories. Evaluations of the systems are given, with a description of the evaluation criteria. A list of nine additional systems is included to direct possible continuation of this effort.

RECOMMENDATIONS

Use the evaluations in this report to direct research into the application of artificial intelligence techniques to command and control, and to focus transition efforts. If additional system evaluations are done, examine the suggested additional systems first.

CONTENTS

1 Introduction

This paper presents a very brief examination of several existing artificial intelligence systems which have been judged to be worth further investigation for use in future command and control applications. Rather than developing all support software from ground zero, it is desirable to make use of existing material and avoid reinventing the wheel.

This document represents the current state of an ongoing process which began with a much longer list of artificial intelligence systems. Each system on the long list was given a very quick evaluation on scales measuring availability, applicability, and maturity. Within each broad category of artificial intelligence effort (e.g. natural language, inference) the highest ranking systems were selected for further examination. This is the result of that examination. The original list and evaluations are included as section 7 of this report.

This is a working document. It is anticipated that some systems were missed and that some evaluations (particularly availability) may be incorrect. Comments, recommendations, and suggested additions are welcomed.

1.1 Availability

High availability ratings indicate that the system is readily available in usable form, usually on a compatible host on the ARPAnet. Evaluations were lowered for systems on incompatible computers, systems which were old enough to probably have disappeared, systems for which documentation or descriptions were lacking, and systems which would require substantial support systems (e.g. speech processing).

Unless otherwise noted, all systems examined run in INTERLISP-10 on a DEC-10 or DEC-20 that is a host on the ARPAnet.

1.2 Applicability

Applicability was judged primarily by evidence of previous work in areas related or similar to command and control. Often, such evidence was not available. When there was no evidence to draw upon, the underlying principles were relied upon for indicators of applicability. This approach resulted in rating most systems "possibly applicable," which made applicability useless as a fine-grained discriminator. However, some systems (like chess players) were clearly inapplicable.

1

1.3 Maturity

Systems whose working principles and theoretical limits are well understood by most workers in the field are highly rated in maturity. Systems understood only by their creators or still in early design stages are rated low for maturity. Most systems fall somewhere in between, with most workers understanding the approach, a smaller set being conversant with the principles of operation, and a very few, if any, able to define the theoretical and practical limitations of the system.

2 Knowledge Representation Systems

Intelligent systems or programs rely on two sources for their power. The first is well designed algorithms and heuristics, which specify the processes to be undertaken in pursuit of a task. The second is accurate, well organized information about the task domain, which provides material for the algorithms and heuristics to work with. The command and control task domain has a wealth of information associated with it, but to date very little of this information has been made available in a useful form for artificial intelligence programs. Organization of the information is as important as its capture, because poorly organized data will either (at best) slow the system or (at worst) be inaccessible.

Knowledge representation systems seek to provide frameworks for organizing and storing information. Designers of different systems perceive different problem areas that need work, and thus different systems do different things well. The goal must be to evaluate the strong and weak points of various representation systems along with the requirements imposed by the information present in the command and control task domain, so that any system selected will be capable of effective performance in the task area, even if it is weak in other areas.

Command and control as a problem domain presents several challenges for knowledge representation systems. Two critical factors in C2 -- time and space -- were considered inadequately understood in a recent survey of workers in the field [1]. A knowledge representation for C2 tasks must have some method for coping with these problems as well as the problems of inheritance (e.g. making sure that all Kynda class platforms have the same maximum speed capability) and synergy (e.g. 25 MIG fighters

---

[1] SIGART Newsletter, Special Issue on Knowledge Representation, edited by RJ Brachman and BC Smith. (70):1-138, February, 1980.

acting together under a unified command can inflict more damage than 25 single fighter raids).

Since these issues must be faced, it is reasonable to consider in greater depth those representation systems which have addressed some of these issues before considering those which have not. The systems considered here each include some form of inheritance capability. Also, these representation systems appear to have sufficient flexibility to permit the embedding of other systems for special-purpose representation, such as for time and space.

2.1 UNITS

UNITS [2] is a package of LISP functions developed at Stanford for creating, deleting, accessing, updating, and otherwise manipulating knowledge representations organized as partitioned semantic networks. In UNITS, the nodes of the network are called units and the links among them are called slots. Units are connected in a generalization hierarchy which permits several modes of property inheritance. The package is available in INTERLISP on the ARPAnet.

Partitioned semantic networks have been proposed as a method of extending the semantic network approach to permit quantification and other desirable features which are not obvious in most "standard" semantic network implementations. The UNITS package is oriented toward this framework for knowledge representation and thus includes any shortcomings that the framework has. Specifically, the criticisms of Brachman concerning the "level" of the representation [3] are valid remarks concerning UNITS.

On the other hand, UNITS has included some features beyond partitioning which many semantic networks do not have. It is possible to attach a procedure to a slot so that the procedure will be executed when appropriate conditions are met -- for example, when the slot is filled.

---

[2] An Examination of a Frame-Structured Representation System, by M Stefik. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pages 845-852. Tokyo, 1979.

[3] On the Epistemological Status of Semantic Networks, by RJ Brachman. BBN Report 3807, Bolt, Beranek, and Newman, April, 1978.

The inheritance hierarchy and defined inheritance modes also impose some much-needed structure on an often nebulous area.

The most attractive feature of the UNITS system is the completeness of the support function library, which indicates a certain maturity. Not only are the "base" network manipulation functions provided, but there is also an editor which knows about slots and units and their structure and a package for displaying networks, which makes it much easier to tell what has been added or changed.

In addition, the designers of UNITS have recognized that many applications will require quite large networks, often bigger than can be readily maintained in core. To support such large networks, a "paging" facility has been provided so that users need not concern themselves directly with the management of memory space.

To date, there have been no efforts to represent naval domains in UNITS, but the similarities between it and KL-ONE are promising.

2.2 KL-ONE

KL-ONE [4] is a package of LISP functions developed at BBN for creating and working with a particular form of associative network representation called a structured inheritance net. As in every associative (semantic) network, the basic entities are nodes and links. KL-ONE shares with UNITS the idea of "typed" nodes, where nodes are required to be of certain limited classes. Also like UNITS, the type of a node constrains the links (slots in UNITS) that can join it to other nodes. However, UNITS permits the user to define new slots that can also be used to link nodes and applies its type restrictions only on inheritance-related links. KL-ONE, by contrast, provides a base or primitive set of links along with the defined node types and does not permit the definition of new links. While this refusal to permit new links may impede flexibility, the claim is that links are always well-defined and never subject to the whims of users. KL-ONE is also available on the ARPAnet in INTERLISP.

KL-ONE was designed in an attempt to provide an epistemologically explicit representation system, with the designer's and user's assumptions open to observation in

[4] KLONE Reference Manual, by RJ Brachman, E Ciccarelli, NR Greenfeld, and MD Yonke. BBN Report 3848, Bolt, Beranek, and Newman, July, 1978.

representations built using the system. Two central aspects of
objects which were felt to be essential to a successful
representation paradigm were the inheritance of properties and
the description of structure. UNITS also addresses the problem
of inheritance but provides no specific facilities for structural
description or specification.

KL-ONE gains particular appeal from its use as a foundation
for other systems, notably the AIPS system [5]. The AIPS work has
used KL-ONE to represent at least simple naval domain
information. The fact that another application with apparently
high relevance has found it useful is certainly encouraging.
Moreover, in using KL-ONE, the AIPS developers have devised some
very useful support functions that make the creation and
examination of large networks much easier. Thus, KL-ONE is
beginning to approach the ease of user interface that is so
promising in UNITS.

2.3 KNOBS

KNOBS [6] is an integrated collection of artificial
intelligence programs directed toward the development of
experimental tactical air command and control. Developed at
MITRE, KNOBS exists in INTERLISP on an ARPAnet host. While there
are several components to KNOBS, it is classified under knowledge
representation systems because that is where the majority of the
effort to date has been concentrated. Building on the framework
of FRL, the KNOBS developers have translated the representation
system into INTERLISP and have augmented it to permit interactive
frame instantiation. This insures that related items, such as the
type of aircraft chosen for a mission and their armament, are
consistent.

KNOBS is clearly oriented toward military applications. For
this reason, there is much in it that is directly applicable to
naval command and control. It is also an integrated system,
combining a powerful knowledge representation with a natural
language interface and some simple inferential capability.

---

[5] Application of Symbolic Processing to Command and Control,
Final Report, by F Zdybel, MD Yonke, and NR Greenfeld. BBN
Report 3849, Bolt, Beranek, and Newman, November, 1979.

[6] KNOBS: An Experimental Knowledge Based Tactical Air Mission
Planning System and a Rule Based Aircraft Identification
Simulation Facility, by C Engelman, CH Berg, and M Bischoff. In
Proceedings of the Sixth International Joint Conference on
Artificial Intelligence, pages 247-249. Tokyo, 1979.

Future plans include the addition of a rule-based inference system. While the level of capability and integration is admirable, KNOBS is still very much a system under development and as such is subject to at least minor change over the near term.

3 Knowledge Presentation Systems

Artificial intelligence programs, as noted earlier, usually require large amounts of information about the domain in which they operate. In addition to this domain knowledge, there is usually also a large amount of information within the domain which the program will process to perform its function. For example, a tactical situation assessment program's domain knowledge may include some formats (representations) for storing information about platforms, sensors, and sightings, while the information that it actually uses in doing assessment concerns real platforms and sightings by sensors, organized as directed by the representations.

Users should have access to the information within the domain that is used by the program, both because it may be useful in raw form and as a check on the program's operation. Managing the presentation of such information is a complex task which has not been as well explored as the problems of information acquisition.

The most widely used technique of knowledge presentation to date has been ordinary text. Occasionally the presentation is organized in a question-answering form, but more commonly it is not under user control at all. It has been especially difficult for users to tailor the information presented to match their needs, concerns, and preferences. Presentation modes other than text have also been extremely limited.

In command and control applications, flexibility to match presentation to needs is essential, as information requirements vary with the situation. Different users may wish to vary the presentation to emphasize points that each considers important, while a single user may wish to vary the presentation over time to satisfy his current information requirements.

Concentration on readily reconfigurable graphics-based presentation systems would be particularly appropriate for C2, where the tool (graphics) can be well matched to the domain (spatial information). Another possible presentation mode that could demonstrate relevance would be synthesized speech output, to utilize a different sensory channel which may be more readily available in some situations.

3.1 AIPS

AIPS (Advanced Information Presentation System) [7] is an intelligent graphics interface designed at BBN especially to support command and control decision tasks. It utilizes representations of knowledge about the domains of graphics, command and control, the display device, and the user's needs to provide a highly interactive, individually tailored graphic display subsystem for use by other systems in the command and control domain. The underlying system for the various knowledge representations is the representation system KL-ONE.

AIPS is available on the ARPAnet, but presumes the availability of a particular bit-mapped graphics display device for successful operation. It does not use GL2 for graphic communication.

Graphics is one of the most useful modes of information presentation and one of the most difficult to manage. The difficulty in use arises from both the complexity of the graphics domain and the lack of standardization among output devices, both of which can impose a heavy programming burden on those who choose to incorporate graphic capability in their systems. AIPS holds out the promise of lifting the graphics interface burden from the developers of other systems, while also making possible graphic presentation specialized for user needs. As mentioned before, some naval-domain-related work has been done in AIPS.

3.2 SDMS

The Spatial Data Management System (SDMS) [8], while not strictly an artificial intelligence system, is an interesting approach to information presentation, particularly of data which would usually be maintained in a traditional database system. SDMS was developed by Computer Corporation of America and is available as an experimental product. Instead of accessing information through a formal query language, SDMS presents the information graphically in a form which seems to encourage browsing and which requires less prior knowledge of the database contents and organization. Data are organized and retrieved by

---

[7] AIPS: An Information Presentation System for Decision Makers by NR Greenfeld and MD Yonke. BBN Report 4228, Bolt Beranek and Newman Inc., December, 1979.

[8] A Prototype Spatial Data Management System by CF Herot, R Carling, M Friedell, and D Kramlich. In SIGGRAPH 80 Conference Proceedings, pages 63-70, 1980.

positioning them in a Graphical Data Space which is viewed through a color raster scan display, permitting both surface traversals and zooms for greater detail.

SDMS is oriented toward the presentation of relatively static, slowly changing information of the type found in a traditional database. It permits an easily learned, highly interactive user interface to such static data but appears to have no provisions for data addition and update, other than through a somewhat laborious redefinition process. While some naval domain data are certainly sufficiently static for this to be a useful system (as shown by the demonstrations in the description), many more data are highly dynamic and so probably not well suited by this particular system. In addition, the system is highly biased toward human use and requires a duplication of the internal database to permit access by both programs and the user.

4 Inference Systems

Inference is the process of drawing conclusions, of adding information to a knowledge (data) base on the basis of information that is already there. Inference may be deductive or inductive, and a given system may permit both forms. Inference systems may operate in many different ways. One of the most useful forms is that of rule-based systems. Here, knowledge is structured in rules which are applied to the facts to reach conclusions. The method of rule application forms the process base for inference; while the rules are, in some sense, the knowledge-structuring base.

Most of the tasks in the command and control domain require some form of inference, of drawing conclusions from known facts. Again, tactical situation assessment is a good example. From (possibly limited) low-level information like sensor reports, it is necessary to reach conclusions about the existence and identity of platforms in the vicinity.

An inference system for C2 tasks must be capable of dealing with information common to the C2 domain. In practice, this will often mean that the inference system must be adaptable enough to work in or with the knowledge representation framework chosen for a task. Flexibility with respect to knowledge representation therefore becomes a major criterion in evaluating systems.

Another important aspect is the extensibility of the inference system. This is one area in which rule-based systems are particularly attractive because, in general, new rules mean new inferences. Among rule-based systems, ease of rule addition and support of the addition function are important.

4.1 E-MYCIN

The original MYCIN [9] was a rule-based system developed at Stanford for diagnosing bacterial infections of the blood. E-MYCIN [10] is essentially the same system, but with the domain-specific information like the rules removed. E-MYCIN operates in a backward-chaining fashion -- given a goal, or statement to be concluded, it applies rules in an attempt to find supporting evidence for the goal. This approach is also called goal-driven inference. It is available on the ARPAnet.

The representation available for rules in E-MYCIN is reasonably sophisticated, and there is a collection of functions which permit a form of reasoning under uncertainty. A related rule addition facility (called TEIRESIAS [11]) may also be available, but the work was done some time ago and may have disappeared. Descriptions of TEIRESIAS are available. However, the available representation for facts is fairly sparse, especially in comparison to systems like UNITS and KL-ONE. Also, it is not clear that a synthesis of systems to overcome this deficiency is possible, and such an inability would impose strict limits on the applicability of E-MYCIN.

Though several problem domains other than blood infections have been investigated using E-MYCIN, none of these involved a naval domain. Most applications have been in medicine.

[9] Computer-Based Medical Consultations: MYCIN, by EH Shortliffe. American Elsevier, 1976.

[10] A Domain-Independent Production-Rule System for Consultation Programs, by W van Melle. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pages 923-925, 1979.

[11] Knowledge Acquisition in Rule-Based Systems: Knowledge about Representation as a Basis for System Construction and Maintenance, by R Davis. In DA Waterman and F Hayes-Roth (editors), Pattern-Directed Inference Systems, Academic Press, 1978.

4.2 ROSIE

ROSIE [12] is another rule-based system which was developed at Rand as a successor to RITA [13], the Rand Intelligent Terminal Agent. ROSIE applies its rules in a forward-chaining manner, going from facts to conclusions. This makes it data-driven rather than goal-driven. The underlying knowledge representation is a common object-attribute-value structure, which has been used in many applications. ROSIE is available on the ARPAnet. A recent major revision has resulted in a system quite different from that available when this survey was started.

A design document exists for the original version. It details a number of features which were not implemented in that version. Some of these features were added in the recent revision. The structures available for facts appear richer than those of E-MYCIN, though they still fall short of the more sophisticated knowledge representation systems (e.g. inheritance mechanisms are not provided, though presumably the user could add them). A synthesis of ROSIE and a more general knowledge representation system might not be as difficult. NOSC personnel have used ROSIE (both in its original incarnation and as revised) to implement rule-based inference systems in naval domains, so applicability is high. However, these same implementations have revealed some awkwardness forced on the user due, at least in part, to the gaps among ROSIE design, implementation, and documentation as well as sparseness of knowledge representation. Past experience does indicate that some command and control tasks are well suited to a data-driven approach.

4.3 STAMMER

STAMMER [14] is a rule-based inference system developed at the Naval Ocean Systems Center for use in tactical situation assessment. Like ROSIE, STAMMER is a forward-chaining system. The

---

[12] Design of a Rule-Oriented System for Implementing Expertise, by DA Waterman, RH Anderson, F Hayes-Roth, P Klahr, G Martins, and SJ Rosenschein. Rand Note N-1158-1-ARPA, The Rand Corporation, May, 1979.

[13] RITA Reference Manual, by RH Anderson, M Gallegos, JJ Gillogly, RB Greenberg, and R Villanueva. Technical Report R-1808-ARPA, The Rand Corporation, 1977.

[14] STAMMER2 Production System for Tactical Situation Assessment, by DC McCall, PH Morris, DF Kibler, and RJ Bechtel. TD 298, Naval Ocean Systems Center, San Diego, CA, October, 1979.

knowledge representation used is a collection of independent assertions, with no structuring mechanism provided in the representation itself. The rule interpreter implements a suspension mechanism for improved time efficiency in repeated rule applications. STAMMER is available on the ARPAnet.

The STAMMER system has been used in three experimental applications to date. The first of these, merchant detection, motivated the addition of graphic display support to the basic system. STAMMER has a moderately sophisticated user interface, with explanation and data base query capabilities among the actions available. Its applicability to tasks in the naval domain is proven, though it is somewhat biased toward the situation assessment task and is not an "empty" system. The lack of facilities for structuring the database is also a problem.

5 Natural Language Processing

The ability to use a natural language such as English to communicate with a computer has long been a goal of artificial intelligence researchers. A language understanding and generating capability could conceivably remove many obstacles that presently obstruct the human-machine interface. Natural language processing, as examined here, is restricted to printed input and output, with speech recognition and generation falling under a different heading.

Natural language processing (NLP) is relevant to command and control in at least two ways. First, much of the present data that needs to be captured for use by an automated C2 system are passed in natural language form (e.g. as "free text" fields in messages). NLP could assist in the capture of this information. Second, NLP could be used to improve the interface between the automated system and its users, making the interface friendlier and more responsive.

One great problem with existing NLP systems is their restrictions on vocabulary and subject area. These restrictions grow from the need to provide domain knowledge to aid in language understanding. For command and control, these restrictions are a benefit. The command and control domain is restricted, and it is reasonable to limit interactions to topics within the domain. This indicates that a limited vocabulary may well be sufficient, so that what would be problems in applying NLP to some other domain may be lessened in command and control.

5.1 ATNs

Recursive augmented transition network parsers (ATNs) have been studied and discussed in the literature for several years. The work that brought them to prominence was done by Woods and others at BBN [15,16]. These parsers work by performing state transitions depending on the value of the word they are presented. The possible transitions form a network, which may be entered recursively. Performing a transition may cause some function to be executed, possibly changing the value of one of the registers which augments the transition net. ATNs have the theoretical power of Turing machines. ATNs are readily accessible. About the worst that could be expected would be a translation from some other LISP dialect. The problem with using an ATN is the writing of grammar and support routines, which will have to be done regardless of the source of the parser itself.

ATNs are very well understood. They have been studied and discussed in the literature for several years. Further, many ATNs are semistandardized, making grammars written for one usually adaptable to another. Even when an existing ATN is not available, implementing one and providing a grammar is almost a cookbook problem [17].

ATNs tend to rely heavily on syntactic features, since those are more readily codified than, say, semantics. In theory, ATNs are not biased toward any particular aspect of language, but in practice most implementations are extended with semantic and pragmatic routines that lie outside the parser proper. The flexibility of ATNs over problem domains has been well demonstrated. However, providing a grammar and any extensions to naval domains would be a user responsibility.

---

[15] Transition Network Grammars for Natural Language Analysis, by W Woods. Communications of the ACM 13:591-606, 1970.

[16] The LUNAR Sciences Natural Language Information System: Final Report, by WA Woods, RM Kaplan, and BL Nash-Webber. BBN Report 2378, Bolt Beranek and Newman Inc., 1972.

[17] Artificial Intelligence Programming, by E Charniak, CK Riesbeck, and DV McDermott. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

5.2 Yale work

Over the past few years, a great deal of work by Roger Schank and his colleagues has come out of Yale in natural language processing [18,19,20,21,22]. The central thesis of most of these efforts has been that language processing is an anticipatory activity, that language is understood by continually making predictions and then testing those predictions against the actual input.

Also, this group has worked on providing conceptual frameworks for language constructs larger than single words or phrases to account for observed features of human language-understanding behavior. The Yale work exists primarily off the ARPAnet in the UCILISP dialect, which would require transport and translation. Work using the same principles has been done or duplicated elsewhere (e.g. CMU) and so might be more available than the original Yale work.

Since the Yale approach to NLP is far different from that of most other workers, comparisons are very difficult. Only during the past year did a study of the differences between a Yale-designed parser and an ATN finally describe the theoretical distinctions.

The Yale work is very interesting in its strong emphasis on domain knowledge to guide understanding. However, this imposes a large burden of domain analysis and encoding on anyone who

---

[18] Analyzing English Noun Groups for their Conceptual Content, by AV Gershman. Research Report 110, Department of Computer Science, Yale University, May, 1977.

[19] Adaptive Understanding: Correcting erroneous inferences, by RH Granger. Research Report 171, Department of Computer Science, Yale University, January, 1980.

[20] Computational Understanding: Analysis of sentences and context, by CK Riesbeck. Working Paper, Fondazione Dalle Molle per gli studi linguistici e di communicazione internazionale, Castagnola, Switzerland, 1974.

[21] Scripts, Plans, Goals, and Understanding: An inquiry into human knowledge structures, by R Schank and R Abelson. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[22] Understanding Goal-Based Stories, by R Wilensky. Research Report 140, Department of Computer Science, Yale University, September, 1978.

attempts to use the system in a new domain. A preliminary study using the general techniques advocated by the Yale group for parsing free text fields in formatted messages was completed in 1979 and indicated evident applicability.

6 Planning and Problem Solving

Planning and problem solving -- the process of determining, examining, and deciding among alternatives -- is at the heart of the command and control domain. Knowledge representation and presentation, natural language interfaces, and inference systems are all useful as components to support the assessment and decision processes.

Current artificial intelligence planning systems combine aspects of the preceding areas to propose action sequences for accomplishing goals. Most of the existing systems also have some ability to monitor the execution of proposed action sequences and to modify the plan adaptively to insure satisfactory achievement of the goal.

C2 planning systems could be used to support decisionmaking by providing an independent source of possible courses of action, accompanying justifications, and assistance in monitoring the execution of selected courses of action. Planning in the C2 environment obviously requires extensive use of domain-specific knowledge such as that cast in the chosen knowledge representation. Ability to work with a powerful representation is a desirable feature. More important is the ability to cope with planning conflicts such as unsatisfied preconditions, limited resources, and counterplanning by other forces. Where possible, a C2 planning system should be able to "learn" common sequences of actions for coping with problems of these types so that costly replanning need not be performed for commonly encountered problems. Execution monitoring is also a very desirable feature.

6.1 STRIPS, ABSTRIPS, and NOAH

STRIPS [23], ABSTRIPS [24], and NOAH [25] comprise a related

---

[23] STRIPS: A new approach to the application of theorem proving to problem solving, by RE Fikes and NJ Nilsson. Artificial Intelligence 2:189-208, 1971.

[24] Planning in a Hierarchy of Abstraction Spaces, by ED Sacerdoti. Artificial Intelligence 5:115-135, 1974.

[25] A Structure for Plans and Behavior, by ED Sacerdoti. Elsevier, 1977.

14

group of planning systems developed at SRI (and Stanford), originally for planning robot movement sequences. STRIPS, the original system, used a predicate calculus representation of the world and the actions available to the robot. Finding a sequence of actions that achieved a goal was recast in this formalism to finding a proof (in the sense of logic) of the goal state description from the initial state description, by using the action descriptions as theorems. Theorem proving techniques, notably resolution, were utilized. Unfortunately, this approach was subject to strong effects from combinatorial explosion, when the size of the theorem-proving space grew much faster than the length of the desired solution path.

A later modification to STRIPS permitted a form of learning in which action sequences that were commonly used could be saved and later used in planning as though they were single operations. The analogy with derived theorems as opposed to axioms in a logic system is very strong.

ABSTRIPS introduced another improvement to the basic STRIPS system, based on the idea of problem abstraction. Operators in STRIPS had preconditions, a set of predicates which were required to be true before the operator could be applied. By ignoring preconditions which were less "critical," ABSTRIPS could perform a quick check to rule out many misleading action sequences before considering fine levels of detail at greater computational cost.

NOAH, while in the broad tradition of STRIPS and ABSTRIPS, is not a direct descendant of the other systems. The interesting idea here involves a procedural net, which links actions or operators flexibly. In many earlier systems, selecting operators committed the system to their order of execution. By representing the actions as nodes in a network whose links are sequencing information, NOAH makes reordering of actions possible without complete replanning.

The availability of these systems is not known. A STRIPS-like system would be relatively easy to implement, ABSTRIPS more difficult, and NOAH relatively complex. Good descriptions of all systems exist in the literature.

The SRI systems have virtually defined the area of planning for artificial intelligence systems. Any planning system must be able to deal with the problems that have been successfully faced by these systems and should attempt to tackle the problems that are still open.

With the possible exception of NOAH, all the SRI systems rely heavily on predicate calculus and theorem-proving techniques. However, the underlying concepts are generally well understood and documented, so adapting the approaches to other representations and inference engines is not impossible.

These planning systems have been used in a variety of domains, though never (so far as is known) in naval areas. Reliance on predicate calculus gives a uniform representation, so that if naval tasks could be expressed in that way, presumably these systems could then do planning in a naval domain. A preliminary study effort is underway to examine the applicability of these approaches.

6.2 TALESPIN

A somewhat different approach to planning is embodied in the TALESPIN [26] program, out of Yale and UC Irvine. While many underlying concepts -- such as preconditions, actions, and postconditions -- are shared with the SRI work, the motivation is quite different, and some critical implementation issues are also approached differently.

TALESPIN is an effort to model (crudely) the planning processes that people use in problem solving. One constraint imposed by the model of people used by TALESPIN is that different representations be used for different kinds of knowledge; thus different inference mechanisms are required. This is in contrast to the SRI approach which uses a uniform representation (predicate calculus) and inference mechanism (theorem proving). TALESPIN, like other Yale work, is in UCILISP on a non-ARPAnet host. Transfer and translation are required.

The vastly different outlooks represented by TALESPIN and the SRI systems make meaningful comparison of the systems extremely difficult. From some viewpoints they appear identical, while from others they are radically different. TALESPIN has not been used for any sort of naval domain problem solving to date. However, its insistence on representations and inference mechanisms tailored to the information involved could prove to be a great aid in developing a naval-oriented system. TALESPIN shares much philosophy and representation with the Yale natural language efforts, which are promising. On the other hand, many of the planning problems addressed by the SRI systems have not yet been considered in TALESPIN, most commonly because they have not arisen in the course of TALESPIN's planning. A current study is examining the TALESPIN approach and the naval domain for applicability.

_____

[26] The Metanovel: Telling Stories by Computer, by JR Meehan. PhD thesis, Yale University, December, 1976.

16

6.3 Other SRI Systems

Since the development of the various STRIPS systems and NOAH, work has continued at SRI in the area of planning. Current efforts include (1) developing a representation for plans and actions that will ease user interaction for interactive plan development [27], and (2) an attempt to perform planning in a community of actors [28], which involves modelling the knowledge and actions of others as well as yourself. Most of this work has just started and is still somewhat in a design phase. However, such systems as have been constructed are presumably on the ARPAnet.

This work addresses two issues that will play an important role in any command and control application of AI planning techniques. Support of user interaction and direction is critical, as is consideration of the plans and actions of others. However, it would seem that neither of these efforts is yet at a useful product stage. Further, they both rely on current SRI representation techniques, which may differ from those used in other naval applications. SRI personnel have expressed interest in using naval domains for planning.

[27] Representing Knowledge in an Interactive Planner, by AE Robinson and DE Wilkins. In Proceedings of the First Annual National Conference on Artificial Intelligence, pages 148-150, 1980.

[28] Multiple-Agent Planning Systems, by K Konolige and N Nilsson. In Proceedings of the First Annual National Conference on Artificial Intelligence, pages 138-141, 1980.

7 Original List and Preliminary Evaluations

Languages

System	Developed by	Grading*
micro-PLANNER	MIT	BBA
CONNIVER	MIT	BBA
SCHEME	MIT	BBB
PROLOG	Waterloo	BBA-
FOLISP	Stanford	BCB
PLASMA	MIT	CB-B
ETHER	MIT	CB-C

Knowledge representation systems (including some other "languages")

UNITS	Stanford	AB+A-
KL-ONE	BBN	AB+A-
KRL	XEROX-PARC	CBB
FRL	MIT	BBA-
KNOBS	MITRE	AA-B
SNePS	Buffalo	B-BB+
Hendrix	SRI	?BB

Knowledge management systems - somewhat different than knowledge representation, though there is certainly considerable overlap. Roughly analogous to DBMS vs. file design.

KBMS	Stanford	?B-B-
KLAUS	SRI	CBB-
CYRUS	Yale	BB-B

Knowledge presentation systems - once you know something, how do you share the knowledge?

AIPS	BBN	A-AB
SDMS	CCA	ABA-

Inference systems (primarily - some also address other issues)

E-MYCIN	Stanford	A-B-A-
Hearsay	ISI(??)	?B+A-
PROSPECTOR	SRI	?BA
ROSIE	RAND	AA-A-
RITA	RAND	ACA
CSA	Maryland	BBB

---

*See next page.

18

Natural Language Processing

ATNs	Everywhere	ABA
LIFER	SRI	B-BA
Yale work (ELI, etc.)		BA-A-
PARSIFAL	MIT	?BB+

Planning and problem-solving

STRIPS	SRI	B-BA
ABSTRIPS	SRI	B-BA
NOAH	SRI	B-BA
BUILD	MIT	?BB
HACKER	MIT	?BB
GPS	CMU	?B-A
TALESPIN	Yale/UCI	B+A-B
Other systems	SRI	BB+B-

Distributed AI and problem solving

Contract nets	MIT,DREA	BBB
FA/C	U Mass	B-BB
DSN	Everywhere	B-BB-

Database interfaces

LADDER	SRI	ABA
TED	SRI	AB+B
TEAM	SRI	CB+C

Miscellaneous

Speech work	various	C+A-B
BKG	CMU	?CA-
CHESS 5.0	NWU	CCA-
Vision work	various	?CB
Theorem proving	various	BBB+

*An explanation of grading:

Availability   ---XXX---   Maturity
                  |
          Applicability

Availability: A - On the net, in a usable form.
                    Trivial acquisition.
                B - Some minor problems; for example,
                    not on net, or wrong LISP dialect.
                C - Enough problems to be an
                    influence on decisions.

19

Applicability: A - Obviously applicable.
                B - Probably useful, but not sure.
                C - Inapplicable.

Maturity: A - Mature. Techniques are well understood by
               a majority of practitioners. Limitations
               as well as benefits are known.
           B - "Adolescence"--generally known but with
               important gaps in knowledge, especially
               in demarcation of powers and limits.
           C - "Infancy"--approach is new; little if
               any established theory or past practice.

These scales are, it is hoped, orthogonal, and no judgment should be based on any one (except for elimination of inapplicable techniques). For example, a nearly mature technique which is hard to acquire may be less useful than a less developed, more accessible one. Also, a high score doesn't imply "goodness." A technology that can be shown to do only one thing can be very mature.

8 Other Systems Suggested for Inclusion

Inference/Representation

KAS/Prospector	SRI
RLL	Stanford
AGE	Stanford
OPS5	CMU
HEARSAY	ISI
EXPERT	Rutgers
HASP	SCI

Knowledge presentation

GUIDON	Stanford

Database interfaces

FQL	UPenn

9 Conclusions

The sixteen artificial intelligence systems summarized here were felt to have some possible relevance to command and control after a preliminary survey. Deeper analysis has borne out this presumption, though it has also shown that many of the systems are still in early research stages and cannot be considered fully mature. Most of the systems have not been applied to specific problems in the Navy domain, so the evaluation of relevance is still preliminary.

10 Recommendations

The summaries of systems in this report should be used as introductions for workers seeking to apply unfamiliar artificial intelligence technology to Navy problems. The evaluations should be used to direct research into the application of this technology to command and control and to focus transition efforts. Additional system evaluations should examine the other systems suggested before seeking out new systems.

END

DATE
FILMED

11-81

DTIC